pyOpenNFT: an open-source Python framework for ML-based real-time fMRI and EEG-fMRI neurofeedback

Ekaterina Antipushina[1*], Nikita Davydov[2*], Riccardo De Feo[1], Evgeny Prilepin[2], Artem Nikonorov[2], and Yury Koush[1✉]

[1] Skolkovo Institute of Science and Technology, Center for Neurobiology and Brain Rehabilitation, Russia
[2] Samara National Research University, Russia
✉yurykoush@gmail.com

Abstract. Real-time functional magnetic resonance imaging (rt-fMRI) is a powerful neuroimaging tool for monitoring brain activity and neurofeedback (NF) applications with promising therapeutic potential in psychiatric and neurological disorders. However, technical implementation of NF using acquired real-time fMRI and/or predicted real-time fMRI signals based on electroencephalographic (EEG) records remains restrictive and often lacks reproducibility. Here, a fully Python-based pyOpenNFT framework was designed for greater flexibility, modularity, and real-time processing efficiency. Its functionality was also extended with a ML-based prediction server for the fMRI NF signal using processed EEG records. The framework streamlines fMRI data acquisition and/or EEG-based prediction, NF signal estimation, and quality assessment (rtQA) without necessarily requiring a GUI. The FastAPI-based implementation for an EEG-based predictor integrates a Lab Streaming Layer (LSL) interface for processed EEG records and delivers real-time predictions of fMRI time-series for target brain regions. The system supports the visualization of additional NF sources by querying a RESTful interface, facilitating interoperability with external applications. Efficient real-time processing is achieved through parallelized workflows, optimized data handling, and shared memory buffers for seamless exchange of brain volumes, time-series data, and rtQA metrics. With open-source code available on GitHub, pyOpenNFT advances multimodal real-time neuroimaging and extends the platform for scientific, clinical and educational applications.

Keywords: real-time fMRI · neurofeedback · pyOpenNFT · real-time quality assessment · brain-computer interface · Python · ML.

1 Introduction

Real-time functional magnetic resonance imaging (rt-fMRI) has emerged as a cornerstone in neuroscience, enabling transformative applications in quantita-

[*] These authors contributed equally to this work.

tive assessment of brain activity and neurofeedback (NF). Recent advancements in MRI hardware, software, and computational methods have further enhanced the precision and utility of rt-fMRI, solidifying its role as an indispensable tool in both research and clinical settings [16]. By leveraging blood oxygen level-dependent (BOLD) signals from whole-brain data and localized regions-of-the-interest (ROIs) in real-time, fMRI allows for monitoring and modulation of target brain activity. This capability has significantly advanced our understanding of brain-behavior relationships and facilitated the development of non-invasive therapeutic interventions for various clinical conditions [16,18,17,19]. However, conventional NF studies face challenges such as consistency, technical complexity, and reproducibility [16,13]. To address these challenges we built upon the open source software OpenNFT for rt-fMRI NF training and quality assessment [6,7,2]. Beyond setting MATLAB aside in favor of a fully open source Python implementation, we introduced pyOpenNFT with significant optimizations for real-time data processing, and machine learning (ML) applications. While fMRI offers in-depth insights into the functional activity and connectivity of the entire brain including deep structures, its R1:[restrictive] experimental setup and high expenses hinder its use for extensive therapeutic settings requiring multiple training sessions. One potential solution to address this challenge is to predict localized fMRI activity based on significantly less expensive and more comfortable EEG records with lower spatial but higher temporal resolution and a model that integrates prior EEG and fMRI data [10,12,8]. This model can then be employed during EEG NF training sessions conducted in natural environments to prevent stress-induced psychopathology [5] and has been shown feasible for borderline personality disorder [20] and post-traumatic stress disorder [4]. Several ML models have been used to predict individual localized fMRI activity responses during EEG-based NF training, which allows a reasonable compensation for the relatively low spatial resolution of EEG records with fMRI priors. Specifically, Meir-Hasson et al. (2014) [12] introduced a ridge regression model to map EEG data to fMRI activity, enabling real-time predictions during NF training. Leibovitz et al. (2021) [9] advanced this approach by combining Long Short-Term Memory (LSTM) networks with Convolutional Neural Networks (CNNs) to capture spatiotemporal patterns in EEG data for more accurate fMRI activity prediction. Semenkov et al. created a lightweight EEG-to-fMRI neural network that predicts detailed BOLD signals, ensuring computational efficiency for real-time applications [14]. Finally, Cury et al. (2020) proposed a sparse regression model that exploits EEG signals alone to predict NF-fMRI or NF-EEG-fMRI activity, reducing the need for concurrent fMRI recordings and enhancing scalability [1]. These advancements underscore the transformative potential of integrating ML into NF applications, paving the way for more effective and personalized therapeutic treatments. The pyOpenNFT framework supports the integration of similar predictors through a FastAPI implementation using RESTful interface and a Lab Streaming Layer (LSL) inlet, enabling EEG data streaming and fMRI NF signal prediction in real-time. For prediction model, we exemplified a basic ridge regression model [12], which could be extended through a user-specific implementation. In summary, our primary contributions include:

- A fully Python-based framework for real-time fMRI data processing. We introduce a comprehensive tool that integrates data acquisition, volume and signal processing, ensuring ease of deployment and reproducibility.

- Efficient data handling and modular architecture. By leveraging shared memory for large-scale data transfers and adopting a modular design, pyOpenNFT supports (non-)GUI modes of operation.

- An interface for EEG-fMRI NF signal prediction. We extended the pyOpenNFT applicability for multimodal neuroimaging studies interfacing the preprocessed EEG data, enabling NF signal prediction using EEG-fMRI ML models.

2 Materials and methods

The pyOpenNFT framework is an open-source software designed for real-time fMRI NF training (Fig.1). The software offers a wide range of functionalities for real-time fMRI studies, such as monitoring fMRI data in real-time, performing standard whole-brain and time-series data processing tasks (e.g., realignment, reslicing, smoothing, incremental general linear model - iGLM, filtering), estimation and presentation of NF signals. pyOpenNFT can generate NF signals in various forms, including target activity levels, functional connectivity (based on single or multiple regions), effective connectivity (using dynamic causal modeling), and classification (via pre-trained support vector machine classifiers). Further developing this functionality, pyOpenNFT distinguishes itself as a fully Python-native evolution of the earlier OpenNFT framework, which relied on hybrid Python-MATLAB interoperability. Its Python implementation relies on free software and was developed to be fully modular, allowing it to be adaptive for specific user cases and workflows. Specifically for core functions like 2D-3D volume mapping, real-time preprocessing, and interprocess communication via memory-mapped files, pyOpenNFT streamlines its architecture using dedicated Python classes and modular processes. Critical operations such as signal preprocessing, volume alignment, and iGLM calculations are now executed natively in Python, eliminating MATLAB dependencies while retaining computational rigor. The interprocess communication is modernized through shared memory buffers and dictionaries, enhancing efficiency for real-time workflows like connectivity analysis or Support Vector Machines (SVM) classification. The redesigned manager-process model centralizes control, allowing users to dynamically enable or disable components (e.g., GUI, data acquisition) without disrupting NF data processing loops. This Python-centric approach not only simplifies installation via standard tools like Poetry but also broadens accessibility, enabling platform-agnostic deployment of advanced fMRI neurofeedback protocols from activity-level monitoring to dynamic causal modeling within a unified, maintainable codebase.

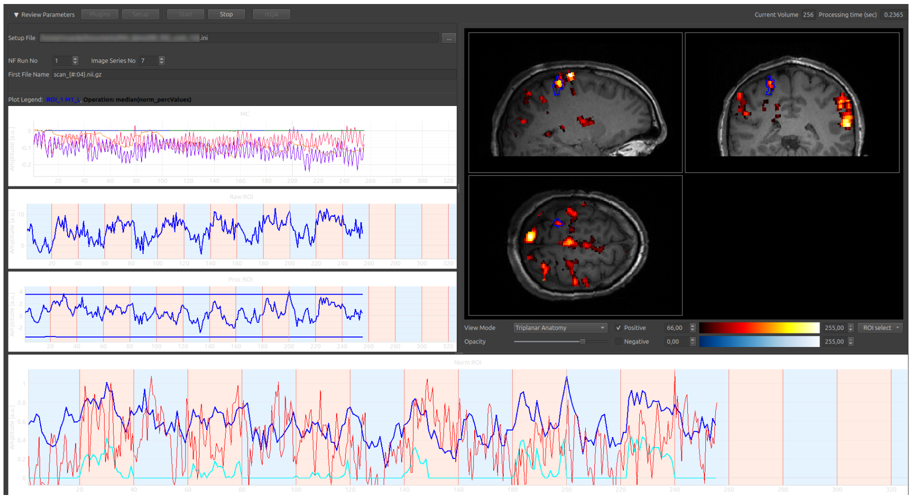

Fig. 1. GUI of pyOpenNFT in a process of a neurofeedback (NF) run, showing EEG-predicted raw fMRI signal (red), as well as raw, filtered and scaled fMRI-based NF signal (blue) from the primary motor cortex. The light blue line represents the displayed NF values derived from the fMRI-based NF signal. Blue and red background areas indicate baseline and regulation blocks, respectively. The right panel displays orthogonal views of the participant's structural MRI, overlaid with the fMRI activation map (iGLM-based) and ROI masks (blue contour).

2.1 pyOpenNFT implementation and design

The pyOpenNFT framework is implemented as a set of communicating Python processes using shared memory buffers for volume and time-series data and dictionaries. It consists of three key processes (Fig. 2): Manager process, Core process, and rtQA process. Each process is implemented as a separate Python class. There are two main processes in the framework – Manager and Core. The Manager process initializes all shared memory buffers, dictionaries, GUI and starts other processes. It loads experimental settings, including the NF study- and session-specific work, watch, structural data and realignment template folders, and the experimental protocol and data acquisition-specific configuration files. The Manager process transfers all necessary information about the experiment via exchange data dictionary to the Core process. The Core process initializes objects of Session and Iteration classes, acquires, processes and transfers volume data between other processes and optionally saves data upon completion of data analyses and at the end of the NF runs. The Session class contains information from configuration file, head motion correction template and target ROIs and methods necessary for experiment' setup and initialization. The Iteration class encapsulates and facilitates processing of the currently obtained volume, time-series and contains the necessary parameters for each step of data processing. The Core Process sequentially monitors the Watch Folder for new data files and performs key volume and time-series data processing operations, including spatial realignment, reslicing, and smoothing, as well as target time-series filtering and feedback estimation. Neurofeedback values could be sent to external software via UDP/COM ports and visualized. Inter-process communication utilizes a periodically checked flag system. GUI plots and images update upon data readiness. At the end of the experimental session, required parameters and results are stored. The rtQA process supports real-time quality assessment for neurofeedback, task-related, and resting-state fMRI paradigms, significantly extending pyOpenNFT functionality. It allows a flexible access to time-series and whole-brain fMRI data at different stages of (pre)processing. For instance, volumetric quality parameters, e.g. whole-brain temporal Signal-to-Noise Ratio (tSNR) or Contrast-To-Noise Ratio (tCNR) could be estimated for raw, realigned, and smoothed volumes, as well as for raw, iGLM-filtered and despiked time-series. Notably, the rtQA extension supports a fully automated mode and could be configured for a particular paradigm. The Image formation helper and rtQA processes are optional and could be disabled in the .yaml configuration file, e.g. in non-GUI and non-rtQA modes, leaving the essential inter-process exchange data dictionary. The Image formation helper process supports visualization of statistical and/or rtQA information in GUI using 2D mosaic or orthogonal 3D volume projections overlaid over motion correction fMRI template and individual structural volumes. For statistical results, positive and negative whole-brain activation maps are computed using iGLM. The non-GUI mode is optionally defined by Manager process suggested as a starting point for customized modification for highest performance NF training that could potentially target minimal feedback delays and large data volumes.

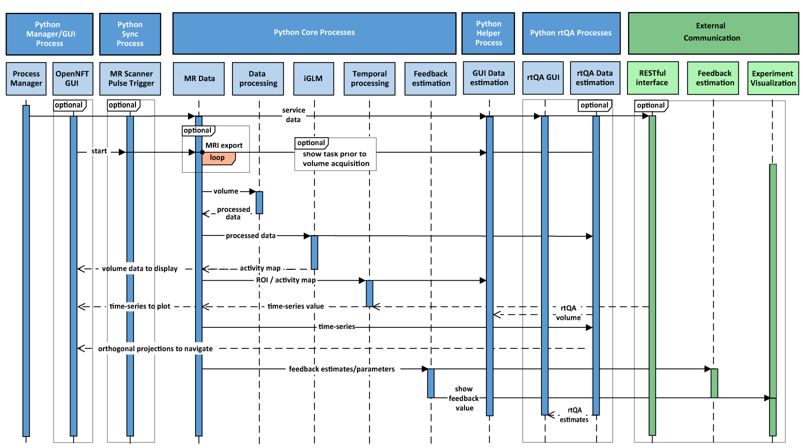

Fig. 2. The pyOpenNFT architecture and workflow.

2.2 Functionality asset and availability

The pyOpenNFT framework consists of several key components enabling real-time fMRI data processing and NF estimation.

nftsession initializes session parameters based on a .ini file and loads the experimental protocol from a .json file.

volume data processing library (`github.com/OpenNFT/python-rtspm`), written in Python and C++, performs the fMRI volume realignment, reslicing, and smoothing.

filewatcher watches for new real-time data monitoring file system events using the watchdog library, forms a file queue and provides the latest available filename.

nftiteration manages the current iteration of data processing, encapsulates the current volume (mrvol) and time-series (mrtimeseries) data, serves as the primary loop handler.

mrvol performs volume acquisition, (pre)processing and iGLM calculation, including spatial realignment, reslicing and smoothing. Of note, we found that MATLAB and Python implementations of linear algebra packages can produce different results for Cholesky decomposition of ill-conditioned matrices, however the detailed investigation of these differences is outside the scopes of the present research. To achieve the higher similarity for iGLM calculations in MATLAB and Python, we assembled the NumPy package and Math Kernel Library (MKL) from Intel. This NumPy+MKL package is installed from our server during installation as specified in .toml file.

mrtimeseries performs time series acquisition and processing, which includes - cumulative GLM using NumPy+MKL package, autoregressive filtering of the first order, time-series denoising and spike correction using modified Kalman filter [8].

nfbcalc performs NF estimation, which includes continuous, intermittent and trial-based types using different activity and connectivity models [6]. The most recent pyOpenNFT version is freely available under the GNU GPL license at GitHub (`github.com/OpenNFT/pyOpenNFT`). Detailed installation instructions and supporting materials are provided at GitHub directories and pyOpenNFT website. The framework uses Poetry for dependency management as specified in pyproject.toml. It supports but is not limited to the core numerical libraries (numpy, scipy), neuroimaging (nibabel, pydicom), and workflow utilities (loguru, watchdog). We use OpenCV to facilitate visualization of ROIs, PyQt6 for GUI implementation, and volume data processing library [2]. Cross-platform compatibility is maintained through integration testing on Windows (11 23H2), and Linux systems using Python 3.10+ environments.

2.3 Neurofeedback estimation based on real-time EEG records using an open multimodal EEG-fMRI dataset

For the prediction of the fMRI signal from EEG data, we selected the method presented by Meir-Hasson et al. [11,12], because of its current utilization in a variety of recent NF studies[15,5]. We exemplify our approach by providing a regularized regressor trained to predict the fMRI signal from the primary motor cortex in openly-available NF data [10]. The regressor was trained on a cluster of EEG-fMRI runs selected for their similarity, based on the parameters characterizing a set of individual regressors trained on each run, resulting in a highly-predictable cluster of localized brain activity (Pearson's r=0.45) [11]. For an in-depth description and evaluation of our implementation of the method, including choice of metrics and alignment between the two modalities, we refer the reader to De Feo et al. [3]. The FastAPI implementation for an EEG-based predictor integrates LSL interface to process EEG records and deliver real-time predictions of fMRI time-series for target brain regions (Fig. 3). The Main thread orchestrates request handling and resource allocation, the LSL listener interfaces with the EEG stream for data ingestion, and parallelized Workers execute computationally intensive tasks (i.e. spectral analysis and feature extraction). To handle high-dimensional neural data efficiently, the architecture employs parallelized workflows, including a dedicated process pool for multi-channel EEG processing. The RESTful interface facilitates HTTP requests through the API and their responses. Specifically, the system constantly collects EEG data through the LSL inlet. When a request is received, and consistently with the original implementation of the method [11,12], the data undergoes spectral decomposition via a Stockwell transform, followed by feature extraction and prediction using a machine-learning model. pyOpenNFT includes a RESTful interface to request predictions and dynamically visualize neurofeedback (NF) sources from external applications.

While we utilize this implementation of a predictor to exemplify a typical workflow, this can be replaced with any alternative user-provided scikit-learn estimator exported by using joblib library. Furthermore, the EEG data processing for inference could be altered by modifying the relevant classes according to the user's needs.

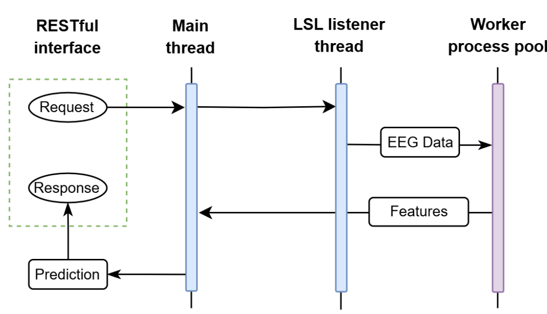

Fig. 3. Overview of the prediction server.

3 Simulation results

For evaluation of pyOpenNFT performance, we used retrospective data from a continuous classification-based NF study [7,2]. We simulated a real-time setting by coping fMRI scans at an interval of TR to the Watch folder. This dataset consists of 210 EPI scans (TR = 2s, TE = 28ms, iso 2.2 mm^3 voxels, volume size = 100 × 100 × 35). The performance was tested on a PC with Windows 11 (Intel Core i9-14900HX, 32 GB RAM) and Linux Mint (AMD Ryzen 9 7950X3D, 128 GB RAM) using real-time simulation of DICOM images (Table 1). For demonstration of EEG-based fMRI prediction, we used continuous EEG NF data [10]. The dataset consists of 332 EPI scans (TR = 2s, TE = 23ms, 2 × 2 × 4 mm^3 voxels, volume size = 210 × 210 × 32). The performance was tested on a PC with Linux Mint using real-time simulation of DICOM images and processed 32-channel EEG data at 250Hz sampling frequency (Table 1; Channel C1, Pearson's correlation with the target ROI of the EEG-based prediction 0.28). The Pearson coefficient obtained in this simulation differs from those reported in the original evaluation of the regressors, because while we can evaluate our approach on open data, this data did not include the mask of target ROI in the same space as the fMRI volumes. As a result, our attempt to reconstruct this ROI independently likely led to a domain mismatch between the current setting and the regressor trained on the signals provided by Lioi et al [10].

Table 1. Computation time comparison across platforms and NF software versions with and without prediction server for EEG-based NF (mean ± std, ms).

OS/Hardware	NF Software	Volume Acquisition	Volume Processing	Time-Series Processing
Windows 11, i9-13900K	pyOpenNFT	1.6 ± 3.7	196.4 ± 22.9	2.4 ± 4.7
	OpenNFT	3.2 ± 0.9	241.5 ± 14.8	7.1 ± 1.4
Linux Mint, Ryzen 9 7950X	pyOpenNFT	3.2 ± 0.9	257.9 ± 28.2	1.4 ± 1.6
	pyOpenNFT and prediction API*	4.3 ± 1.5[1]	221.1 ± 23.2[1]	1.9 ± 1.7[1] 0.08 ± 0.02[2]

* Denotes the total EEG-based real-time fMRI prediction time.
[1,2] Denote fMRI and predictor processing time.

4 Conclusion

Neurofeedback based on real-time fMRI is a promising therapeutic approach with significant potential for treating psychiatric and neurological disorders.However, limitations of fMRI in cost, portability, and practical implementation highlight the need for complementary approaches. Electroencephalography (EEG) emerges as a portable, cost-effective alternative, ideal for ecological settings and longitudinal training, which however suffers from relatively low spatial resolution and inability to accurately access brain activity of deep brain structures. The use of multimodal EEG-fMRI records prior to EEG-based NF training and ML models that predict fMRI signals in real time from EEG records can provide the balanced NF signal with relatively high temporal accuracy of EEG and high spatial resolution of fMRI. This expands the scope of NF therapy and makes it more scalable and practical for real-world clinical applications, bridging the gap between research and therapeutic applications. The pyOpenNFT framework targets versatile real-time fMRI(-EEG) data processing and feedback estimation operations in Python, offering high modularity and accessibility for neuroimaging research. It balances between research, clinical and educational aspects of the open-source software, which aligns with modern approaches for scalability in neural datasets. The unification of the parallel architecture based on optimized inter-process communication and data flows enables further extensions of software operation modes and functionality. This places pyOpenNFT in a prominent position among the choices for real-time fMRI processing software.

Acknowledgments. AN and ND were supported by World-Class Research Center of Intelligent UAS (XX XXX XX).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Cury, C., Maurel, P., Gribonval, R., Barillot, C.: A sparse eeg-informed fmri model for hybrid eeg-fmri neurofeedback prediction. Frontiers in neuroscience **13**, 1451 (2020)
2. Davydov, N., Peek, L., Auer, T., Prilepin, E., Gninenko, N., Van De Ville, D., Nikonorov, A., Koush, Y.: Real-time and recursive estimators for functional mri quality assessment. Neuroinformatics **20**(4), 897–917 (2022)
3. De Feo, R., Antipushina, E., Tyukin, I., Koush, Y.: Eeg-to-fmri prediction for neurofeedback: Evaluating regularized regression and clustering approaches. bioRxiv (2025). https://doi.org/10.1101/2025.05.20.654907, https://www.biorxiv.org/content/early/2025/05/26/2025.05.20.654907
4. Fruchtman-Steinbok, T., Keynan, J.N., Cohen, A., Jaljuli, I., Mermelstein, S., Drori, G., Routledge, E., Krasnoshtein, M., Playle, R., Linden, D.E., et al.: Amygdala electrical-finger-print (amygefp) neurofeedback guided by individually-tailored trauma script for post-traumatic stress disorder: Proof-of-concept. NeuroImage: Clinical **32**, 102859 (2021)
5. Keynan, J.N., Cohen, A., Jackont, G., Green, N., Goldway, N., Davidov, A., Meir-Hasson, Y., Raz, G., Intrator, N., Fruchter, E., et al.: Electrical fingerprint of the amygdala guides neurofeedback training for stress resilience. Nature human behaviour **3**(1), 63–73 (2019)

6. Koush, Y., Ashburner, J., Prilepin, E., Sladky, R., Zeidman, P., Bibikov, S., Scharnowski, F., Nikonorov, A., Van De Ville, D.: Opennft: An open-source python/matlab framework for real-time fmri neurofeedback training based on activity, connectivity and multivariate pattern analysis. NeuroImage **156**, 489–503 (2017)
7. Koush, Y., Ashburner, J., Prilepin, E., Sladky, R., Zeidman, P., Bibikov, S., Scharnowski, F., Nikonorov, A., Van De Ville, D.: Real-time fmri data for testing opennft functionality. Data in brief **14**, 344–347 (2017)
8. Koush, Y., Zvyagintsev, M., Dyck, M., Mathiak, K.A., Mathiak, K.: Signal quality and bayesian signal processing in neurofeedback based on real-time fmri. NeuroImage **59**(1), 478–489 (2012)
9. Leibovitz, R., Osin, J., Wolf, L., Gurevitch, G., Hendler, T.: fmri neurofeedback learning patterns are predictive of personal and clinical traits. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 282–294. Springer (2022)
10. Lioi, G., Cury, C., Perronnet, L., Mano, M., Bannier, E., Lécuyer, A., Barillot, C.: Simultaneous eeg-fmri during a neurofeedback task, a brain imaging dataset for multimodal data integration. Scientific data **7**(1), 173 (2020)
11. Meir-Hasson, Y., Keynan, J.N., Kinreich, S., Jackont, G., Cohen, A., Podlipsky-Klovatch, I., Hendler, T., Intrator, N.: One-class fmri-inspired eeg model for self-regulation training. PLoS One **11**(5), e0154968 (2016)
12. Meir-Hasson, Y., Kinreich, S., Podlipsky, I., Hendler, T., Intrator, N.: An eeg finger-print of fmri deep regional activation. Neuroimage **102**, 128–141 (2014)
13. Ros, T., Enriquez-Geppert, S., Zotev, V., Young, K.D., Wood, G., Whitfield-Gabrieli, S., Wan, F., Vuilleumier, P., Vialatte, F., Van De Ville, D., et al.: Consensus on the reporting and experimental design of clinical and cognitive-behavioural neurofeedback studies (cred-nf checklist) (2020)
14. Semenkov, I., Rudych, P., Ossadtchi, A.: Beyond the surface: Revealing the depths of brain activity by predicting fmri from eeg with deep learning. bioRxiv pp. 2024–11 (2024)
15. Singer, N., Poker, G., Dunsky-Moran, N., Nemni, S., Balter, S.R., Doron, M., Baker, T., Dagher, A., Zatorre, R.J., Hendler, T.: Development and validation of an fmri-informed eeg model of reward-related ventral striatum activation. Neuroimage **276**, 120183 (2023)
16. Sitaram, R., Ros, T., Stoeckel, L., Haller, S., Scharnowski, F., Lewis-Peacock, J., Weiskopf, N., Blefari, M.L., Rana, M., Oblak, E., et al.: Closed-loop brain training: the science of neurofeedback. Nature Reviews Neuroscience **18**(2), 86–100 (2017)
17. Sorger, B., Goebel, R.: Real-time fmri for brain-computer interfacing. Handbook of clinical neurology **168**, 289–302 (2020)
18. Stoeckel, L., Garrison, K.A., Ghosh, S.S., Wighton, P., Hanlon, C.A., Gilman, J.M., Greer, S., Turk-Browne, N.B., deBettencourt, M.T., Scheinost, D., et al.: Optimizing real time fmri neurofeedback for therapeutic discovery and development. NeuroImage: Clinical **5**, 245–255 (2014)
19. Tschentscher, N., Tafelmaier, J.C., Woll, C.F., Pogarell, O., Maywald, M., Vierl, L., Breitenstein, K., Karch, S.: The clinical impact of real-time fmri neurofeedback on emotion regulation: A systematic review. Brain Sciences **14**(7), 700 (2024)
20. Zopfs, M., Jindrová, M., Gurevitch, G., Keynan, J.N., Hendler, T., Baumeister, S., Aggensteiner, P.M., Cornelisse, S., Brandeis, D., Schmahl, C., et al.: Amygdala-related electrical fingerprint is modulated with neurofeedback training and correlates with deep-brain activation: proof-of-concept in borderline personality disorder. Psychological Medicine **54**(8), 1651–1660 (2024)